# Video Analysis API: A Comprehensive Guide

Welcome to our Video Analysis API documentation. This guide will walk you through the intricacies of our system, which leverages artificial intelligence to analyze video content and respond to natural language queries. Let's dive in and explore how this API can revolutionize your approach to video content analysis.

## Table of Contents

1. [API Overview](#api-overview)

2. [Authentication](#authentication)

3. [Endpoints](#endpoints)

4. [Testing with Postman](#testing-with-postman)

5. [Architecture Deep Dive](#architecture-deep-dive)

## API Overview

Our Video Analysis API is a robust system designed to transform unstructured video content into queryable data. It combines several key technologies:

- Django-based RESTful API

- JWT (JSON Web Token) authentication

- AI-powered transcript analysis

- Vector embeddings for efficient querying

This integration allows users to register, authenticate, submit videos for analysis, and query the processed content using natural language.

## Authentication

Authentication is a crucial aspect of our API, ensuring secure access to its features. We implement JWT (JSON Web Token) for this purpose.

### What is JWT?

JWT is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. In our API, JWTs are used to authenticate users and maintain session state.

### How JWT Works in Our System

1. User logs in with credentials

2. Server validates credentials and issues a JWT

3. Client stores the JWT (typically in local storage)

4. Client includes the JWT in the header of subsequent requests

5. Server validates the JWT for each request to protected endpoints

This stateless authentication method allows our API to scale efficiently while maintaining security.

## Endpoints

Let's explore each endpoint in detail:

### 1. Register User

- \*\*Endpoint\*\*: `/register/`

- \*\*Method\*\*: POST

- \*\*Purpose\*\*: Creates a new user account

\*\*Request Body\*\*:

```json

{

"username": "new\_user",

"email": "user@example.com",

"password": "secure\_password\_123"

}

```

\*\*Response\*\*:

```json

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

```

### 2. Login User

- \*\*Endpoint\*\*: `/login/`

- \*\*Method\*\*: POST

- \*\*Purpose\*\*: Authenticates a user and provides a JWT

\*\*Request Body\*\*:

```json

{

"email": "user@example.com",

"password": "secure\_password\_123"

}

```

\*\*Response\*\*:

```json

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

```

### 3. Refresh Token

- \*\*Endpoint\*\*: `/token/refresh/`

- \*\*Method\*\*: POST

- \*\*Purpose\*\*: Issues a new access token using a valid refresh token

\*\*Request Body\*\*:

```json

{

"refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

```

\*\*Response\*\*:

```json

{

"access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

```

### 4. Analyze Transcript

- \*\*Endpoint\*\*: `/analyze-transcript/`

- \*\*Method\*\*: POST

- \*\*Authentication\*\*: Required (JWT)

- \*\*Purpose\*\*: Submits a video for transcript analysis

\*\*Request Body\*\*:

```json

{

"video\_url": "https://www.youtube.com/watch?v=example\_id"

}

```

\*\*Response\*\*:

```json

{

"message": "Transcript analysis completed",

"video\_id": "example\_id"

}

```

### 5. Query Transcript

- \*\*Endpoint\*\*: `/query/`

- \*\*Method\*\*: POST

- \*\*Authentication\*\*: Required (JWT)

- \*\*Purpose\*\*: Queries the analyzed transcript using natural language

\*\*Request Body\*\*:

```json

{

"video\_id": "example\_id",

"query": "What are the main topics discussed in this video?"

}

```

\*\*Response\*\*:

```json

{

"result": "The main topics discussed are: 1) Introduction to APIs, 2) Authentication methods, 3) Natural language processing in video analysis."

}

```

## Testing with Postman

Postman is an excellent tool for testing APIs. Here's a step-by-step guide to test our endpoints:

1. \*\*Install Postman\*\*: Download from [postman.com](https://www.postman.com/)

2. \*\*Set Up a New Request\*\*:

- Click "New" → "HTTP Request"

- Select the appropriate HTTP method (GET, POST, etc.)

- Enter the endpoint URL (e.g., `http://localhost:8000/register/`)

3. \*\*Configure Request Body\*\*:

- For POST requests, go to the "Body" tab

- Select "raw" and choose "JSON" from the dropdown

- Enter the required JSON payload

4. \*\*Add Authentication\*\*:

- For protected endpoints, go to the "Authorization" tab

- Select "Bearer Token" from the Type dropdown

- Enter your JWT in the Token field

5. \*\*Send the Request\*\*:

- Click "Send" and observe the response

6. \*\*Analyze the Response\*\*:

- Check the status code, headers, and body of the response

- For successful requests, you should see a 200 OK status and the expected JSON response

## Architecture Deep Dive

Our API's architecture is designed for scalability, efficiency, and ease of integration with AI technologies. Let's break down the key components:

1. \*\*Django REST Framework\*\*: Provides a robust foundation for building RESTful APIs, handling routing, serialization, and view logic.

2. \*\*JWT Authentication\*\*: Ensures secure, stateless authentication, allowing the API to scale horizontally without shared session stores.

3. \*\*Transcript Analysis Pipeline\*\*:

- Speech-to-Text Conversion: Transforms video audio into text.

- Text Preprocessing: Cleans and normalizes the transcript text.

- Chunking: Splits large texts into manageable segments.

- Embedding Generation: Creates vector representations of text chunks.

4. \*\*Vector Storage (Pinecone)\*\*: Efficiently stores and indexes text embeddings, enabling fast similarity searches.

5. \*\*Query Processing (LangChain)\*\*:

- Converts natural language queries into vector representations.

- Performs similarity search against stored embeddings.

- Synthesizes relevant information to generate a response.

This architecture allows for efficient processing of large volumes of video content and provides quick, accurate responses to user queries.

By understanding these components, you can leverage the full power of our Video Analysis API, opening up new possibilities in content analysis and information retrieval.

Remember, the key to mastering any API is practice. Start with simple requests, gradually increasing complexity as you become more comfortable with the system. Happy coding!